|  |  |  |
| --- | --- | --- |
|  | Главная страница - AITU | **Outline Proposal Form**  for Design Patterns Courses’ Project Work |
| **Software Design Patterns Courses’ Project Work Review** **November  2023** | | |  |
| Student | | Alamanova Alua, Aubakirova Daniya,  Kaiyr Nurasyl |  |
| Group | | SE-2214 |  |
| Course Instructor | | Mr. Nurtas N.K. |  |
| **Topic of The Project:**  The Topic: “*Manga (Japanese Comic Books) Online shop”*  Working Name: *“Otaku Manga”* | | |  |
| **Description:**  The *"Otaku Manga"* online shop project is a Java-based application designed to provide a user-friendly platform for purchasing Japanese Comic Books. This project serves as an educational exercise in software design patterns, particularly focusing on the implementation of the Strategy, Observer, Decorator, Singleton, State and other 18 design patterns.  Project Features:  Homepage: The project's homepage features navigation buttons, such as Product catalog, Shopping Cart, Order Status, and Confirm Order, providing a straightforward and intuitive user interface through the console.  Product catalog: Within the Category section, various products with specific prices are displayed. Users can add products to their shopping cart, and this functionality is implemented using the Singleton design pattern, ensuring that there is only one shopping cart instance throughout the application.  Shopping Cart: The Shopping Cart allows users to enhance their selected products with additional features like packaging, express delivery, and discounts using the Decorator design pattern. The basic order contains the price of the product from the chosen category. The status of the order is set to "IN PROCESSING" using the Observer and State patterns.  Payment Method: Users can select their preferred payment method, choosing from Credit Card, PayPal, or WebMoney. This is achieved through the implementation of the Strategy design pattern. The order status is updated to "SENT" using the Observer and State patterns to reflect the progress of the order.  Order Status: The "Order Status" section keeps users informed about the status of their orders. The Observer and State patterns are employed to provide real-time updates on order progress.  Order History: the Memento Pattern is used to implement the Order History feature, allowing users to view their previous orders. Class ShoppingCartMemento saves the current state of the shopping cart using the saveOrder() method. And allows users to view their last order by retrieving the last saved state from the order history using the getLastOrder() method.  Order Confirmation: Upon order confirmation, the status is set to "DELIVERED" using the Observer and State patterns to indicate that the order has been successfully delivered to the customer.  Save Order To Txt File: which is a command to save an order to a text file. This class has an execute method that takes a list of products (order history) and writes this information to the specified text file using Command pattern. The Order History class uses an instance of the Save Order To Text File class to execute the save order command.  Cart, History, State: Using Facade pattern simplifies user interactions with the "Otaku Manga" online shop by providing an easy way to add products to the cart, process payments with various methods, and view order history, all while abstracting the underlying complexities. This design pattern enhances the user experience and maintains clean and efficient code.  GUI: Our GUI interface for the ‘Otaku Manga’ store project not only incorporates various design patterns but also offers a user-friendly experience for customers, making it easier to explore and purchase Japanese comics. The seamless interaction between these patterns enhances the overall functionality and aesthetic appeal of our application.  Coding Language: Java  Interface: GUI  Pattern: Strategy, Observer, Decorator, Singleton, State, Memento, Facade, Command patterns  **Expected Result:**   * Creation of diverse object classes * Functioning project * Seamless integration of design patterns * User-friendly Text User Interface (TUI) * Implement additional design patterns * Integration of a comfortable user interface (UI) * Efficient and well-optimized codebase * Adherence to coding standards and best practices * Regular updates and improvements * A project that serves as a valuable learning experience in software design and development | | |  |
| **ER Diagram** | | |  |
| **Progress (Completed)**  Home Page:   * Implemented Product catalog button for product navigation. * Added a Shopping Cart button for managing selected items. * Created an Order Status button to track order progress. * Integrated a Confirm Order button to finalize purchases.   Product catalog:   * Displayed several products with specific prices for user selection. * Implemented the ability to add products to the shopping basket, utilizing the Singleton pattern for cart management.   Basket:   * Enhanced product offerings by allowing users to add various options like packaging, express delivery, and discounts. This was achieved using the Decorator pattern. * The basic order now contains the price of the selected product from the chosen category. * Set the order status to "IN PROCESSING" using the Observer and State patterns.   Payment Method:   * Enabled users to choose their preferred payment method from Credit Card, PayPal, or WebMoney, leveraging the Strategy pattern. * Updated the order status to "SENT" using the Observer and State patterns to reflect the progress of the order.   Order Status:   * Implemented a section to display the current order status. * Utilized the Observer and State patterns to provide real-time updates on the order's progress.   Order History:  - Displaying the last ordered product using Memento Pattern  Order Confirmation:   * Set the order status to "DELIVERED" using the Observer and State patterns upon order confirmation, indicating successful order delivery.   Facade Pattern:   * Created a facade to simplify the complex ordering process by hiding the details of subsystems. * In this facade class, the complex interactions with the subsystems like adding products to the cart, processing payments, and displaying order history are encapsulated.   Save To TXT File:   * Implemented the Command Pattern to enhance the functionality of the "Otaku Manga" online shop project. * Specifically, the Command Pattern is applied to the "Save to Text File" feature for order history. This command allows users to save their order history to a text file for future reference. * By introducing this pattern, the project gains the ability to execute commands that provide additional functionality and improve the overall user experience.   GUI:  - The JavaFX framework was used to create a graphical interface for the project, which made it possible to implement a convenient and intuitive user experience. | | |  |
| **Code/Screen your project** | | |  |